COSC 1104 – Assignment 1

# Learning Goal

This assignment is meant to be a body of work that you produce in Python to both develop and demonstrate your mastery of foundational programming concepts using Python. The solution should make use of several concepts already introduced, including selection, iteration, lists, and functions.

This work is meant to be completed individually and without the use of generative AI. If other web-based resources are used to complete this work, these should be referenced in code comments.

Choose one of the two options below for this assignment:

# Option 1:

Create an application that will analyze whether a number is a prime number or not and provide some additional information around it.

First, you will need to define a function that determines if a number is prime or not. There are many versions of this already; you are encouraged to write this function yourself, but if you use one found on the web please test it thoroughly and cite where it came from.

When the script runs, request input from the user. They should enter a positive whole number. If they enter something that is not a positive whole number, they should be prompted again. There should be no way to cause a crash.

If the user enters a positive whole number, output should include the following:

1. The prime number *before* the number they entered, if there is one.
2. Whether their number is prime.  
   If it is not prime, display a list of the number’s divisors.
3. The next prime number *after the* number they entered.

Sample output:

Please enter the number to check: kyle

That is not a positive whole number. Try again.

Please enter the number to check: 7

The prime number before 7 is 5.

7 is a prime number.

The prime number after 7 is 11.

Press Enter to exit the program...

Please enter the number to check: 12

The prime number before 12 is 11.

12 is not prime. Its factors are 2, 3, 4, 6.

The prime number after 12 is 13.

Press Enter to exit the program...

# Option 2:

This program is based around the creation of a simulated cloud storage tracking system.

You will need to define at least these three functions:

* A function to create a user account. Its parameters will be a username, and the amount of storage space this used has available. When the user account is created, the username will be added to a list of usernames and the storage space will be added to a list for each user’s available space.  
  This should include some validation to ensure the username is unique (and not blank!) and the storage space is a positive number.
* A function representing the upload of a file. Its parameters will be a username, a filename, and a filesize. If the user exists and has enough space to upload the file, update the used space.
* A function to delete a user account.

1. Create a new account.
2. Delete an account.
3. Upload file.
4. List accounts.
5. Exit

Select an option (1-5):

When the program starts, a menu will appear as shown on the right.

When the user selects 1, this should call the function to create a user account. The user will need to enter some information for this to work.  
When the user selects 2, this should call the function to delete a user account.  
When the user selects 3, this should call the function for uploading a file.  
When the user selects 4, this should neatly display all of the accounts that currently exist including their username and their available storage.

This menu should cycle endlessly, without crashing, and only exit on condition that the user enters menu option 5.

# Constraints:

This assignment is meant to emphasize the use of iteration, lists and functions. It should be completed without external libraries or classes; if you feel external libraries or classes are necessary or particularly useful, seek approval from the professor **before** completing the work.

Adhere to your assigned style guide and ensure your filenames, variables, constants and other elements are properly and consistently cased and adequately descriptive.